CLAIMS

What is claimed is:

1. A distributed multiprocessing computer system, which includes a plurality of processors each coupled to an associated memory module, wherein each memory module may store data that is shared between said processors, said system comprising:

a Home processor that includes a memory block and a directory for said memory block in an associated memory module;

an Owner processor that includes a cache memory, and wherein said Owner processor obtains an exclusive copy of said memory block, and stores said exclusive copy of said memory block in said cache memory; and

wherein said Owner processor may displace the exclusive copy of said memory block, and return said displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor remains a sharer of said memory block.

2. The distributed multiprocessing computer system of claim 1, wherein said Owner processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the Home processor indicates that said Owner processor has obtained exclusive control of said memory block.

3. The distributed multiprocessing computer system of claim 2, wherein said Owner processor is capable of executing multiple threads concurrently, and may displace data associated with a non-executing thread from its associated cache memory.

4. The distributed multiprocessing computer system of claim 3, wherein said Owner processor includes a register in which an address is stored representing the memory block obtained in response to the Load Lock instruction, and wherein said Owner processor compares the address of any displaced data with the address stored in said register.

5. The distributed multiprocessing computer system of claim 4, wherein the Owner processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register.

6. The distributed multiprocessing computer system of claim 5, wherein the Owner processor asserts a Victim message if the address of any displaced data does not match the address stored in said register.

7. The distributed multiprocessing computer system of claim 5, wherein the directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

8. The distributed multiprocessing computer system of claim 7, wherein said Owner processor subsequently re-obtains an exclusive copy of said memory block to complete execution of the non-executing thread.

9. The distributed multiprocessing computer system of claim 8, wherein the Owner processor asserts a Read-with-Modify Intent Store Conditional instruction to the Home directory to again request an exclusive copy of said memory block.

10. The distributed multiprocessing computer system of claim 9, wherein, in response to the Read-with-Modify Intent Store Conditional instruction, the Home directory determines if the Owner processor is a sharer of the memory block, and if so, the Home directory sends an exclusive copy of the memory block to the Owner processor.

11. The distributed multiprocessing computer system of claim 10, wherein the Home directory invalidates all other sharers when it sends an exclusive copy of the memory block to the Owner processor.

12. The distributed multiprocessing computer system of claim 9, wherein the Home directory determines if the Owner processor is a sharer of the memory block, and if not, the Home directory sends a Store Conditional Failure message to the Owner processor.

13. A method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors, comprising the acts of:

requesting a copy of a memory block from a Home processor to perform a write operation on the copy of the memory block;

storing said copy of said memory block exclusively in a cache memory associated with an Owner processor;

updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor;

displacing said copy of said memory block from said cache memory prior to completion of operations on said memory block;

transmitting a message to said Home processor relinquishing exclusive control of said memory block, while indicating that said Owner processor should still be deemed a sharer of said memory block.

14. The method of claim 13, wherein the copy of the memory block is requested using a Load Lock instruction from the Owner processor to the Home processor.

15. The method of claim 13, wherein the Load Lock instruction forms part of a Load Lock/Store Conditional instruction pair.

16. The method of claim 13, wherein the act of updating the coherence directory includes modifying a register to indicate that the Owner processor has an exclusive copy of the memory block.

17. The method of claim 13, wherein the act of displacing said copy of said memory block includes comparing the address of any displaced memory block with an address of any memory block for which an exclusive copy resides in the Owner processor.

18. The method of claim 17, wherein the act of transmitting a message includes assertion of a Victim To Shared message if the address of the displaced memory block matches the address of any memory block for which an exclusive copy resides in the Owner processor, and wherein the directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

19. The method of claim 18, further comprising the act of updating the coherence directory to indicate that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

20. The method of claim 13, further comprising the act of asserting a request to again obtain an exclusive copy of said memory block.

21. The method of claim 20, wherein, in response to request to again obtain an exclusive copy of the memory block, the Home processor determines if the Owner processor is a sharer of the memory block, and if so, the Home processor sends an exclusive copy of the memory block to the Owner processor.

22. The method of claim 21, wherein the Home processor invalidates all other sharers when it sends an exclusive copy of the memory block to the Owner processor.

23. The method of claim 20, wherein, in response to request to again obtain an exclusive copy of the memory block, the Home processor directory determines if the Owner processor is a sharer

of the memory block, and if not, the Home directory sends a Store Conditional Failure message to the Owner processor.

24. A distributed multiprocessing computer system, comprising:

a first processor that includes a memory block and a directory associated with said memory block that tracks the status of said memory block;

a second processor that includes a cache memory, and wherein said second processor is capable of requesting an exclusive copy of said memory block that is stored in said cache memory; and

wherein said second processor may displace the exclusive copy of said memory block prior to completing processing of said memory block, and said second processor transmits a signal to said first processor indicating that said second processor relinquishes exclusive control of said memory block but should remain a sharer of said memory block.

25. The distributed multiprocessing computer system of claim 24, wherein said second processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the first processor indicates that said second processor has obtained exclusive control of said memory block.

26. The distributed multiprocessing computer system of claim 24, wherein said second processor is capable of executing multiple threads concurrently, and may displace data associated with a non-executing thread from its associated cache memory.

27. The distributed multiprocessing computer system of claim 24, wherein said second processor includes a register in which an address is stored representing the memory block exclusively obtained from said first processor, and wherein said second processor compares the address of any displaced data with the address stored in said register.

28. The distributed multiprocessing computer system of claim 27, wherein the second processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register.

29. The distributed multiprocessing computer system of claim 28, wherein the directory associated with the first processor indicates that said second processor has become a sharer of said memory block in response to said Victim To Shared message.

30. The distributed multiprocessing computer system of claim 24, wherein said second processor subsequently re-obtains an exclusive copy of said memory block from said first processor to complete processing of said memory block.

31. The distributed multiprocessing computer system of claim 30, wherein the second processor asserts a request to read, modify, and conditionally store said memory block to said first processor.

32. The distributed multiprocessing computer system of claim 31, wherein, in response to the request to read, modify, and conditionally store said memory block, the first processor determines

if the second processor is a sharer of the memory block, and if so, the first processor sends an exclusive copy of the memory block to the second processor.

33. The distributed multiprocessing computer system of claim 32, wherein the first processor invalidates all other copies of said memory block when it sends an exclusive copy of the memory block to the second processor.

34. The distributed multiprocessing computer system of claim 31, wherein, in response to the request to read, modify, and conditionally store said memory block, the first processor determines if the second processor is a sharer of the memory block, and if not, the first processor sends a failure message to the second processor.